Step 5: Rendering items from an external file

# Introduction:

New features in this exercise:

* [props](https://www.w3schools.com/react/react_props.asp)
* [.map](https://www.w3schools.com/jsref/jsref_map.asp)

In this exercise we’re going to use React to render a list of items from an **external JavaScript file**. You don’t have to write the file that works as our database yourself. You can copy the items [here](itemDataBase.js.docx).

In this step we’ll learn to use two new features, called **props** and **map**. Let’s take a look at them first:

What is props?

props is a special keyword in React, which stands for properties and is used for **passing data** from one component to another.

But the important part here is that data with props is passed in a uni-directional flow (**one way from parent to child**).

\*\*\*CHILD COMPONENT\*\*\*

function Fruit(props) {

    return <p>{ props.fruit }</p>;

}

\*\*\*PARENT COMPONENT\*\*\*

function FruitBasket() {

  return (

    <>

      <h1>What fruit is in my basket?</h1>

      <Fruit fruit="Orange" />

    </>

  );

}

props can also be used by replacing the “props” with the word of the **prop itself**. Basically, just removing the “props.” all together:

\*\*\*CHILD COMPONENT\*\*\*

function Fruit(fruit) {

    return <p>{ fruit }</p>;

}

In the example above, you have two components, a **child component** and a **parent component**. The parent component passes data **to** the child component using props. The parent component has a header and under it, renders the child component, **setting** the fruit-prop to orange. The child component renders that fruit in the p tags.

What is map in React?

The map method is heavily used in React. The most common use of map in React is to **render a list** in one way or another. A map is **not** a feature of React. Instead, it is the standard JavaScript function that could be called on an array. The map method creates a new array by calling a provided function on every element in the calling array.

const employees = ["Diana", "Mark", "Kevin"];

export const Employees = () => {

    return (

    <div>

        {employees.map((employee) => (

        <p>{employee.name}</p>

        ))}

    </div>

    );

};

In the example above, you have a very simple array of employees. Instead of having to manually list the employees one by one, it’s using the map method to do so. The map method **goes through** the array and renders every item there, making it very easy to display the names.

# Item.jsx:

Now that we are somewhat familiar with props and map, let’s continue our exercise

Start by creating a file called **itemDatabase.js** under the pages folder and paste the items there. After saving the file, create a new file called **Item.jsx** in your pages folder, this is going to be our **child component**.

Start by creating a base to your component and adding props as the components parameter:

import React from "react";

export const Item = (props) => {

    return (

        <div>

        </div>

    );

}

Our “database” has 4 fields per item:

* Title
* Description
* Image
* ID

We are going to render **the first three** of them, leaving the ID hidden. Let’s create two p tags inside our components div element and one img tag. Name the p tags to **Title** and **Description**:

<div>

     <p>Title: </p>

     <p>Description: </p>

     <img src= />

</div>

Now, we’ll add our props into the tags. We will name them simply **props.<name of the field>**, props must be in **squirrely brackets**:

import React from "react";

export const Item = (props) => {

    return (

        <div>

             <p>Title: {props.title}</p>

             <p>Description: {props.description}</p>

             <img src={props.img}/>

        </div>

    );

}

These are going to act kind of like a “frame” for our fields that we’re going to render. The fields will be inserted **in the place** of the props tags.

# Items.jsx:

Our child component is now ready, let’s make our **parent component**. Inside the pages folder, create a file called **Items.jsx**.

**Import** the items from the itemDatabase.js file and the Item component, then create a base to our component as usual.

import React from "react";

import { Item } from './Item';

import { items } from './itemDatabase';

export const Items = () => {

    return (

        <div>

        </div>

    );

}

Now we’ll use the map method in our code. Let’s start by inserting the items map inside our components div element (**map has to be inside squirrely brackets**):

<div>

    {

        items.map(({}))

    }

</div>

Inside the maps squirrely brackets, we will insert our **item fields** that we want to render, which were Title, Description and Image.

items.map(({title, description, img}))

Now we will insert them into our Item component which is the child component. We will render the Item component in this file and feed it the data we want to render:

<div>

    {

        items.map(({title, description, img}) =>

        <Item

            title={title}

            description={description}

            img={img}

        />)

    }

</div>

Now everything should work, all that’s left to do is adding a route to our Items component in our App.js file and link in our router that we made in the previous exercise:

function App() {

    return (

        <BrowserRouter>

            <div>

                <Routes>

                    <Route path='/' element={<Router />}>

                        <Route index element={<Message />} />

                        <Route path='Buttons' element={<Buttons />} />

                        <Route path='Items' element={<Items />} />

                    </Route>

                </Routes>

            </div>

        </BrowserRouter>

    );

}

export const Router = () => {

  return (

    <>

        <div>

            <ul>

                <li>

                    <Link to="/">Home </Link>

                </li>

                <li>

                    <Link to="/buttons">Buttons</Link>

                </li>

                <li>

                    <Link to="/items">Items</Link>

                </li>

            </ul>

        </div>

        <Outlet />

    </>

  )

};

Now we should have a rendered list of our items in our browser:

Graphical user interface

Description automatically generated(Image size reduced with CSS)

We have now learned the basics of props and mapping in React. In the next exercise we will start our two-part component where we will render **submitted data**. **See you there!** 😊